A.1 Python Built-In Function Reference

Adapted from https://docs.python.org/3/library/functions.html. Note that not all built-in functions are shown.

Built-in Function	Description
`abs(x)`	Return the absolute value of a number. The argument may be an integer or a floating point number.
`all(iterable)`	Return `True` if all elements of the `iterable` are true (or if the `iterable` is empty).
`any(iterable)`	Return `True` if any element of the `iterable` is true. If the `iterable` is empty, return `False`.
`chr(i)`	Return the string representing a character whose Unicode code point is the integer *i*. For example, `chr(97)` returns the string `'a'`, while `chr(8364)` returns the string `'€'`. This is the inverse of `ord()`. The valid range for the argument is from 0 through 1,114,111. `ValueError` will be raised if *i* is outside that range.
`divmod(a, b)`	Take two (non complex) numbers as arguments and return a pair of numbers consisting of their quotient and remainder when using integer division. For integers, the result is the same as `(a // b, a % b)`.

`filter(`
`  function,`
`  iterable`
`)`
	Construct an iterator from those elements of `iterable` for which `function` returns `True`. `iterable` may be either a sequence, a container which supports iteration, or an iterator.

`id(object)`	Return the "identity" of an object. This is an integer which is guaranteed to be unique and constant for this object during its lifetime.

Built-in Function	Description

`input([prompt])`	If the `prompt` argument is present, it is written to standard output without a trailing newline. The function then reads a line from input, converts it to a string (stripping a trailing newline), and returns that. Example:
	
	`>>> s = input('Type in a word: ')`
	`Type in a word: Python  # "Python" is user input`
	`>>> s`
	`'Python'`

`isinstance(`
`  object,`
`  classinfo`
`)`
	Return `True` if the `object` argument is an instance of the `classinfo` argument, or of a subclass thereof. If `object` is not an object of the given type, the function always returns `False`.

`len(s)`	Return the length (the number of items) of an object.

`# Form 1`
 	
`max(`
`  iterable,`
`  *`
`  [, key,`
`  default]`
`)`
 	
`# Form 2`
 	
`max(`
`  arg1,`
`  arg2,`
`  *args,`
`  [key]`
`)`
	Return the largest item in an `iterable` or the largest of two or more arguments.
	
	If one positional argument is provided, it should be an iterable. The largest item in the iterable is returned. If two or more positional arguments are provided, the largest of the positional arguments is returned.
	
	There are two optional keyword-only arguments. The `key` argument specifies a one-argument ordering function like that used for `list.sort()`.
	
	The `default` argument specifies an object to return if the provided iterable is empty. If the iterable is empty and `default` is not provided, a `ValueError` is raised. If multiple items are maximal, the function returns the first one encountered.

Built-in Function	Description

`# Form 1`
 	
`min(`
`  iterable, *`
`  [, key,`
`  default]`
`)`
 	
`# Form 2`
 	
`min(`
`  arg1,`
`  arg2,`
`  *args,`
`  [key]`
`)`
	Return the smallest item in an `iterable` or the smallest of two or more arguments.
	
	If one positional argument is provided, it should be an iterable. The smallest item in the iterable is returned. If two or more positional arguments are provided, the smallest of the positional arguments is returned.
	
	There are two optional keyword-only arguments. The `key` argument specifies a one-argument ordering function like that used for `list.sort()`.
	
	The `default` argument specifies an object to return if the provided iterable is empty. If the iterable is empty and `default` is not provided, a `ValueError` is raised. If multiple items are minimal, the function returns the first one encountered.

`open(`
`  file,`
`  mode='r'`
`)`
	Open `file` and return a corresponding file object. If the file cannot be opened, an `OSError` is raised.
	
	`file` is a path-like object giving the pathname (absolute or relative to the current working directory) of the file to be opened.
	
	`mode` is an optional string that specifies the mode in which the file is opened. It defaults to 'r' which means open for reading in text mode. Other common values are 'w' for writing (truncating the file if it already exists), 'x' for exclusive creation and 'a' for appending.

`ord(c)`	Given a string representing one Unicode character, return an integer representing the Unicode code point of that character. For example, `ord('a')` returns the integer `97` and `ord('€')` (Euro sign) returns `8364`. This is the inverse of `chr()`.

`pow(`
`  base, exp`
`  [, mod]`
`)`
	Return `base` to the power `exp`; if `mod` is present, return `base` to the power `exp`, modulo `mod` (computed more efficiently than `pow(base, exp) % mod`). The two-argument form `pow(base, exp)` is equivalent to using the power operator: `base ** exp`.

`print(`
`  *objects,`
`  sep=' ',`
`  end='\'`
`)`
	Print `objects` to standard output, separated by `sep` and followed by `end`. `sep` and `end`, if present, must be given as keyword arguments.
	
	Both `sep` and `end` must be strings; they can also be `None`, which means to use the default values.

`reversed(seq)`	Return a reverse iterator.

Built-in Function	Description

`round(`
`  number`
`  [, ndigits]`
`)`
	Return `number` rounded to `ndigits` precision after the decimal point. If `ndigits` is omitted or is `None`, it returns the nearest integer to its input.
	
	For the built-in types supporting `round()`, values are rounded to the closest multiple of 10 to the power minus *ndigits*; if two multiples are equally close, rounding is done toward the even choice (so, for example, both `round(0.5)` and `round(-0.5)` are `0`, and `round(1.5)` is 2).
	
	Any integer value is valid for *ndigits* (positive, zero, or negative). The return value is an integer if ndigits is omitted or None. Otherwise the return value has the same type as number. *Note*: The behavior of `round()` for floats can be surprising: for example, `round(2.675, 2)` gives `2.67` instead of the expected `2.68`. This is not a bug: it's a result of the fact that most decimal fractions can't be represented exactly as a float. See Floating Point Arithmetic: Issues and Limitations for more information.

`sorted(`
`  iterable, *,`
`  key=None,`
`  reverse=False`
`)`
	Return a new sorted list from the items in `iterable`.
	
	Has two optional arguments which must be specified as keyword arguments.
	
	`key` specifies a function of one argument that is used to extract a comparison key from each element in iterable`(for example,`key=str.lower`). The default value is`None`(compare the elements directly).`reverse`is a boolean value. If set to`True`, then the list elements are sorted as if each comparison were reversed.

`sum(`
`  iterable,`
`  /,`
`  start=0`
`)`
	Sums `start` and the items of an `iterable` from left to right and returns the total.

`type(object)`	Return the type of an `object`. The `isinstance()` built-in function is recommended for testing the type of an object, because it takes subclasses into account.